再谈一次 AI 的逻辑结构

YKY 甄景贤

July 30, 2020

Table of contents

多谢 支持 ☺️

Curry-Howard isomorphism

- Curry-Howard isomorphism 是现代逻辑的 核心思想，但我初时没有留意，以致很多东西 看不懂，明白之后 豁然开朗
- 它讲的是 逻辑证明 与 计算 之间的深刻关系：

$$\begin{aligned} \text{逻辑} &\Leftrightarrow \text{type theory} \\ \text{逻辑证明} &\Leftrightarrow \text{programs} \end{aligned} \tag{1}$$

- 先看 $\Rightarrow$：逻辑是一些 符号 / 形式上 的推导，所以 逻辑证明 (derivations) 必然对应於某种 计算 (computation)，这很容易理解
- 再看 $\Leftarrow$：当人们企图定义 普遍的计算模式 (models of computation) 时，发觉 总是对应於 某些逻辑；这一点比较神秘，我也不完全了解
- 这个关系的 发现者 至少包括:
  Brouwer-Heyting-Kolmogorov-Curry-de Bruijn-Howard

Type theory

- 首先了解一下 type theory, 它起源於 Bertrand Russell 为了避开 数理逻辑上的 悖论 (paradox) 的尝试
- 后来发现 type thoery 用来定义 programs 很有用
- 大家都知道 Lisp 语言没有 types，它是一种 untyped λ-calculus
- 在 Lisp 之上引入 type system，衍生成 ML, Caml, OCaml, Haskell, 等 一系列的语言
- 每一个 program 属於某个 type，例如 $\mathrm{length}()$ 函数：

$$\mathrm{length} : \mathsf{String} \to \mathsf{Integer} \tag{2}$$

它输入一个 字串，输出一个 整数（字串的长度）

Curry-Howard isomorphism 的一些细节

- 例如 我们定义一个函数 f, 由 A 类 映射到 B 类:

$$f : A \to B \tag{3}$$

- 这对应於 逻辑上「A 蕴涵 B」的关系:

$$A \Rightarrow B \tag{4}$$

- 逻辑上，每个命题 A 都有一个 proof object 或 witness, 记作 $\Box : A$
- 而函数 f 就是将 $a : A$ （A 的证明）映射到 $b : B$ （B 的证明）:

$$f : a \mapsto b \tag{5}$$

- 类似地，有函数可以将 两个分开的命题 A 和 B 的证明 映射到 $A \wedge B$ 的证明，这里不赘述了
- 简言之，可以建立 命题逻辑 的 $\Rightarrow, \wedge, \vee, \neg$ 对应到一些函数上，这些函数用 typed λ-calculus 定义
- 但与 λ-calculus 对应的逻辑 没有 排中律，它是 intuitionistic logic. 这种逻辑 符合 数学上的 构造主义 (constructivism)

谓词逻辑 (predicate logic)

- 刚才说明了，type theory 对应於 命题逻辑
- 如果要处理 predicates，需要一些 特殊的 types
- Predicate 就是一个有「洞」的命题，填入某些 objects 之后变成真正的命题
- 换句话说 predicate P 是一个函数 $P : X \to \mathcal{U}$，对每个 x 产生 $P(x)$，这 $P(x)$ 也是一个 type, 而 $\mathcal{U}$ 是所有 types 的 universe
- 这样会产生一些不是很像「命题」的 types，例如 自然数的类 $\mathbb{N}$, 似乎不是命题；关於这点我暂时仍不太明白
- 重点是：type theory 很自然地 同构於 categories，with objects $=$ types and morphisms $=$ function types $(A \to B)$
- 而在 categories 中，predicate 的结构可以用 fibration 描述，记作 $\begin{array}{c}\mathbf{Pred}\\ \downarrow \\ \mathbf{Set}\end{array}$
- Fibrations 符合某种 universal property，是从代数几何／拓扑 借来的概念

Logic's topology and geometry

- 有些逻辑学家 察觉到 类型论 的 $\tau_1 \to \tau_2$ 和逻辑中 $P \to Q$ 是一模一样的
- 这关系的深刻之处，在於把 符号逻辑上的 proofs 和 程式语言 的 programs 划上等号，前者是 符号 / 静态的，后者是 程序 / 动态的
- 每个 proof 就是一个 program，它输入一些 arguments，输出 关於那些 arguments 的证明
- 例如：「所有人都会死」是一个 program，它输入「苏格拉底」，输出「苏格拉底会死」
- 这个对应也許可以应用到深度学习：神经网络 也是一種 函數 / mapping，它将 逻辑前提 map 到结论
-

和 AI 的关系

- 那既然 AI 基於 逻辑，而逻辑的结构 如上所述，则 AI 与逻辑之间 必然存在 精确 (precise) 的联系
-

逻辑的 invariance 结构

- 以前曾经说过，机器视觉 的成功，有赖於 将 视觉的几何结构 impose 在深度神经网络上
- 这 深度神经网络 原本是 “free” 的，但加了限制之后，权重空间 变小了（例如维数降低），所以学习加速了
- 所谓 symmetry 的意义，简单例子：「如果知道左边等於右边，那就只需计算一次」
- 换句话说，数学家喜欢对称性，是因为它经常可以简化计算
- 同理，我们想将 逻辑结构 的对称性 impose 到神经网络
- 实际上，可能只需要逻辑上的交换律，就可以达到 强人工智能，正如 机器视觉的成功，在於引入了 CNN 的 convolution 结构，后者只是 视觉不变性 的其中一个最显著的 invariant
- 现代逻辑理论 非常漂亮，我花了十多年时间才弄懂，我希望将这套 逻辑-学习 理论简单讲解一下，也算功德完满了

- 在经典时代，逻辑的 代数形式 可以用 Boolean algebra 表述，然而这方法只适用於 命题逻辑
- Boolean algebra 是中学生熟悉的，类似 Venn diagram 的结构
- 这种结构和 拓樸学 的 open sets 结构一样，所以 命题逻辑 也可以看成是一种 topology
- 然而 predicate logic 的结构更复杂，直到最近才有比较完善的表述
- 现代逻辑结构和 type theory 有深刻的关系，此即 Curry-Howard isomorphism
- 现代逻辑也涉及 topos theory，那是一种由 algebraic geometry 引入的结构

Topos theory and fibrations

- Predicate logic （谓词逻辑）和 命题逻辑 之间的差异在於 fibration 结构
- Fibration 通常用 $\begin{array}{c}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{array}$ 表示，$\mathbb{B}$ = base space, $\mathbb{E}$ = ètalè space, p = projection
- Base space 是 type 的空间，ètale space 是 predicate 的空间
- 由於 Curry-Howard 对应, type = propositions，在 base 空间上只有 命题逻辑
- 例如 $\mathbb{B}$ 空间的一个 type 是 Human，$\mathbb{E}$ 空间的一个谓词是 Mortal
- 於是有以下这个 type inference rule:

$$i : \mathrm{Human} \vdash \mathrm{Mortal}(i) : \mathsf{Prop} \tag{6}$$

意思是说，如果 i 属於 Human 类型，则 $\mathrm{Mortal}(i)$ 属於 Prop 类型

- $H(a)$ is a type.
- H is a prodicate type.
- The proof of $H(a)$ may be the tuple (a, H) or $a \in H$

「神经」知识表示

为什么要研究 神经知识表示?

- 从经典 logic-based AI 的传统，一直在使用「符号」的知识表示法
- 符号逻辑 很容易转换成 抽象代数 / 范畴论 形式（它们是同一个大家庭的「近亲」）
- 然而 或许存在 截然不同 的知识表示法? 但我们很难想像它 长什么样子
- 人脑的「神经」知识表示，可以作为参考，然后再研究它和逻辑表示之间的 correspondence

神经知识表示 的特点:

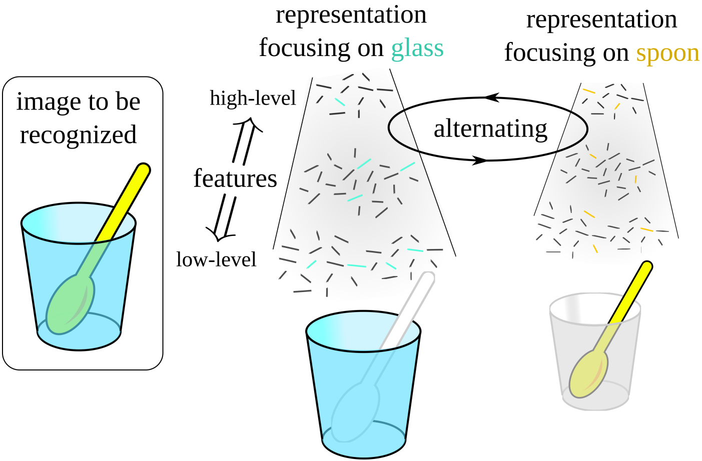
- distributive（分布性）
- model-based (vs rule-based)
- *in situ*（固定性）— 例如辨认「猫」的时候，大脑中 相应的神经元被 激活，但这些 神经元 不能移动，所以「猫」的表示 也不可移动

问题是：如果要辨认「白猫追黑猫」，「猫」的表示是固定的，则这两个「猫」表示 如何共存於神经网络中?

答案很可能是：两个「猫」交替地 出现在 时间上

神经 特征簇 (feature clusters)

例如，以「匙羹在杯中」作例子：

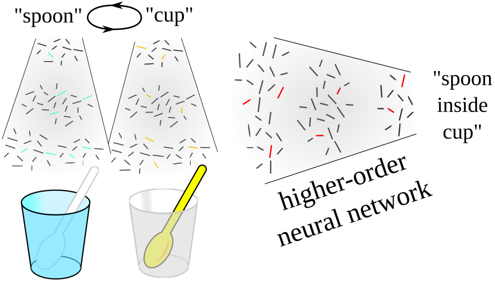


(7)

每个 复杂物体 由一个 feature cluster 辨认。多个「特征簇」在时间上交替出现，可以看成是一种 composition，例如 $A \cdot B$ 或 $A \circ B$.

高阶 特徵

- 一串 特征簇 的时间序列，例如 $A \cdot B$，可以被 更高阶 的神经网络 用作输入。高阶辨认 的结果是一些关系 (relations)，例如「匙羹在杯内」



(8)

- 这似乎是一个 特征空间 $\times$ 时间 的映射 $f : X \times T \to Y$
- 关於这部分其实我仍未肯定，或许有其他方法

关於 “model-based reasoning” 的质疑

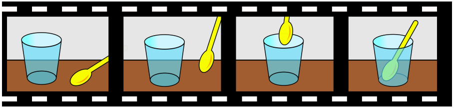
- 很多人认为大脑的思考方式是 先在脑中构造 models，然后再从 models 中「读出」一些结论
- 例如给定一个描述：「已婚妇人出轨，用刀刺死丈夫」

(9)

- 如果假设「妻子有长头发」、「丈夫死时穿著西装」，这些都是 臆想 出来的细节，是不正确的
- 那么这 model 可以有哪些细节？答案是：任何细节都不可以有，除非是 逻辑上蕴含的
- 例如我们可以假设妻子 probably 有一双手臂，但也有例外的情况是独臂的，这是一种 逻辑推导
- 所以，其实所谓 “model-based reasoning” 并没有那么神奇，也并不一定正确，它的细节必需被 逻辑 约束
- 而 model 本身也可以用一些 抽象的逻辑命题 构成，这也是合理的；反而，一个有很多感官细节的 model 并不合理

神经 $\leftrightarrow$ 逻辑 correspondence

- 我们的目标是了解 神经表示 和 逻辑表示 之间的关系，这关系或许可以用范畴论描述?
- 定义 复杂情境 (complex scenario) 是 感知材料 (sensory data) 的一个片段，如:



(10)

又或者一个故事，例如「John 爱 Mary 但 Mary 不爱他」

- 一个复杂情境 可以用若干个 特征簇 描述
- Equivalently, 复杂情境 可以用 逻辑 表示，就是一大堆 逻辑命题 的 conjunction，这些命题 钜细无遗 地描述该情境

多谢收看 ☺️